

The Program Modeling and Simulation of Cellular Function Expression

Jun Ma^{*1}, Mei Fan², Yide Ma³

^a The School of Information Science and Engineering, Lanzhou University
No.222, TianShui Road(south), ChengGuan District, LanZhou City, GanSu Province, China

^{*1}majun@lzu.edu.cn; ²fanmei@lzu.edu.cn; ³ydma@lzu.edu.cn

Abstract

Based on the gene expression process through two stages: transcripts and translation, and such a fact that embryonic stem cell must be division and differentiation, this paper used the reflection-based object-oriented programming skills to simulate the gene expression, cell division and differentiation process. Then through the logic analysis the following hypothesis was proposed: the DNA molecular chain is a program coding sequence, similar to the modern computer program coding sequence that contains abstract data code and corresponding control instruction code. The paper used analogy analysis and simulation method, and the life phenomenon and computer process, genome code sequences and computer program code sequences were compared, then their common properties were abstracted and analyzed, and some computer programs were designed to simulate the cellular function expression and other key procedure. From the simulation results, it can be seen that the genetic code's working mechanism in cells is similar to the Java program working mechanism in computers, and the life phenomena and computer processes are in line with the principle of program (ordered set of instructions) execution. From the perspective of program, the DNA code sequences and the life phenomena were studied to understand the structure of the genetic code and the cell's working mechanism. It was pointed out how to qualitative the relationship of DNA code, RNA code and protein code in the program perspective.

Keywords

Programming Hypothesis on Life Phenomena; Reflection Technology; Gene Expression; Expression of Cell Function; Programming Simulates

Introduction

It has been learned by experiments that the chick hatching process has two necessary conditions: First requires the intact DNA molecular chain, while the second requires appropriate material and energy input. Based on the modern biology-related knowledge, it has been known that the double helix structure DNA molecular chain is composed of the four nucleotides

that have store all the genetic information of the organism(Crick,1958). Analysis and study of the coding sequence formed by ATGC nucleotides are the forefront of Molecular genetics (Stenesh, 1989; Steiner, 1965) and Genomics (Brown, 1999). There are many similarities between DNA coding sequences and computer program coding sequences and consequently the programming explanation has become popular in this field. Richard Dawkins (1995) has indicated that "The machine code of the genes is uncannily computer-like. Apart from differences in jargon, the pages of a molecular biology journal might be interchanged with those of a computer engineering journal." and Bill Gates (1997) believed "Human DNA is like a computer program but far more advanced than any software we've ever created." Using the computer technology to study the coding sequence included in DNA molecular chain is the hot spots in the frontier, meanwhile it is the main method used by bioinformatics (Mount, 2000; Cristianini NaH, 2006). Jun M. et al. (2013) have proposed a new idea and method to study the DNA coding sequences, that is, if the DNA molecular chain is assumed to be a program coding sequence, similar to the modern computer program coding sequence, which contains abstract data code and corresponding control instruction code, then the life phenomena must be running-status's macro performance when its saved-status program coding sequences have been loading and running, similar to the concept of "process" in computer fields.

Through the program modeling and simulation running test, it was found that the life program running mechanism is similar to a virtual machine running mechanism. Examining from the level of code, there are three levels of code in life phenomena: DNA, RNA and protein. It is similar to the bytecode in storing status, in running status and true machine code in the JVM system. The life phenomenon is a "running" process supported by the energy continuous supply.

The Similarity Analysis of the Life Program and the Java Bytecode Program

First of all, it has been well known from the composition of the coding sequence that the DNA code is a combination of the four codes 'ATGC', while the Java bytecode program is a combination of two codes '01'. Secondly, The DNA coding sequence contains introns and exons, and the Java bytecode program coding sequence contains instruction code and data code. Finally, from the transformation process in which the coding sequence in stored status is changed to running status, it can be seen that the life program in cell firstly converts the DNA coding sequences into the RNA coding sequence, namely transcription in biology, then the RNA coding sequences are translated and cut into proteins, this called the translation process in biology, at last the proteins are involved in all sorts of chemical reaction which finally completed the metabolism. Through analysis, it is speculated that the DNA code is a intermediary code, mainly used to store the life program coding sequence and genetic breeding, and not directly involved in the metabolism of chemical reactions, similar to the Java program's executing process. The life program is contrast with Java program execution, and transcription corresponds to the storage stage of bytecode sequences loading into JVM and makes it into running stage, while translation corresponds to the JVM transforming the bytecode sequences into machine code instructions or data code or a native procedure call, finally the metabolism of cells corresponding to the CPU to execute the machine instruction sequences to show the specific procedure function. The comparison of the life program and the Java program is shown in figure 1, in which subgraph **a** illustrates the similarities of Java bytecode codes in computer program and DNA codes in cell program. **b** illustrates the Central Dogma of molecular biology. **c** illustrates the central role of the bytecode sequence which is composed by the JVM instructions and in which **A** indicates that JVM loads bytecode sequence and makes it into running state; **B** indicates that JVM instructions are translated into machine instructions and then executed; **C** indicates that the bytecode instructions are written into the stored bytecode sequence reversely, similar to the process as RNA reverse transcription to DNA; **D** indicates that the bytecode sequence replicates itself which is controlled by machine instructions, similar to DNA replication

process which is controlled by DNA polymerases.

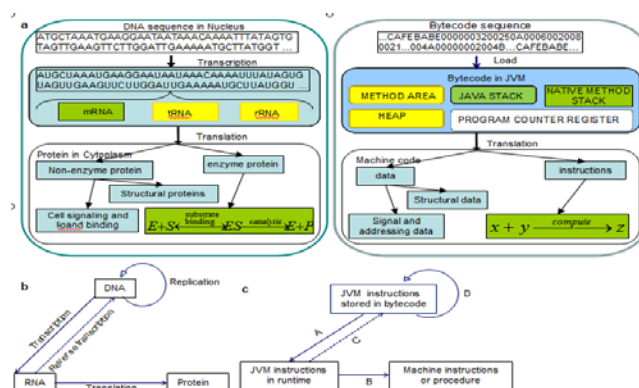


FIGURE 1 THE COMPARISON OF CELL LIFE PROCEDURE AND JAVA PROGRAM EXECUTION PROCEDURE

The Programmed Explanation of the Cell Functional Expression

Since each codon is composed of three nucleotides and these codons are fixed arrangement pattern, so there are $4^3=64$ codons in all. For example, a RNA sequence UAGCAUCC contains three codon: UAG, CAA and UCC, and this RNA codes represents a protein sequence which has three amino acid. DNA sequence is similar, but with T instead of U. Here it has been noticed that this basic codon table is limited in fixed mode, and in the modern computer system, both the basic instructions set and the basic character code set that achieved any program are limited by the combination of a number of bits encoded. Thus, these codes is thought similar to modern computer program binary coding system, four kinds of basic code 'ATGC' encoding a wide variety of life program.

The central dogma of genetic (Watson, 1976) indicates that DNA contains all the genetic information of living systems; in other words, DNA is the source of all genetic information, the only carrier of genetic information to life. And the specific function of the cell is provided by the protein. In the process of making proteins, the genes are from DNA transcription to the corresponding RNA template, namely the messenger RNA (mRNA). Then in the ribosomes and transfer RNA (tRNA) and some action of the enzyme, the template RNA is translated into the amino acid composition of the polypeptide chain, and after the corresponding post-translational modification then the special protein is formed, as shown in Figure 2.

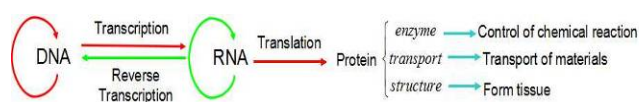


FIGURE 2 DNA AND PROTEIN TRANSLATION

In the life program, it is speculated that the basic instruction set is a variety of enzyme in which one controls a special chemical reaction or a variety of transport proteins, and a particular enzyme or transport proteins corresponds to a instruction of the computer's CPU, such as plant photosynthesis, respiration and amino acids polypeptide chemical equation, etc., as shown in Figure 3:

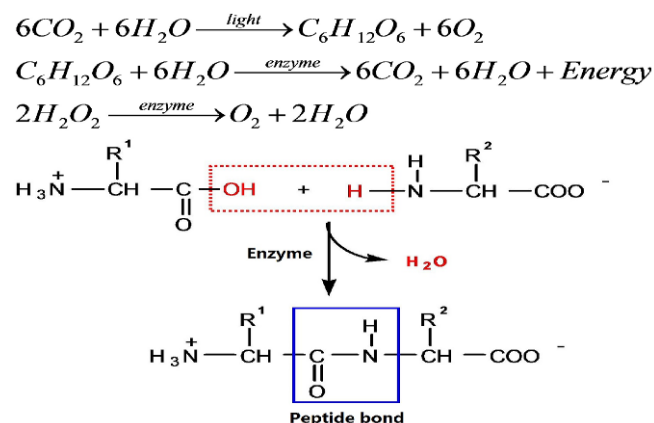


FIGURE 3 PHOTOSYNTHESIS, RESPIRATION AND OTHER CHEMICAL EQUATION

So essentially the life phenomenon is a complex program to maintain the "self-existence". This "self-existence" is a non-material but cannot be separated from the material or energy. It can only be observed macroscopically but not measured microscopically. In this sense, the current part of the research for the life sciences is not feasible, as the most of the current research are aimed at specific gene fragments for local studies. Just like we open a computer program and study a fragment of coding sequence, to observe the relationship between the code fragment and the macro run results(or system status), it seems to get a section of program code corresponding to a special running state, but the whole program and the meaning of existence are failed to be understood.

The specific function of the cell is through specific protein determined by the nucleus of specific gene locus on a chromosome. This process first began in the nucleus, chromosomes in the nucleus begins unwinding, and then the appropriate gene fragments are selected "transcribed" to form mRNA fragments. Further, the mRNA moves into the cytoplasm through the nuclear pore, then in the cytoplasm, making this mRNA as a template, a protein is synthesized which has the particular amino acid sequence in the ribosome, and this process is called "translation" in modern biology. The specific functions and characteristics of a cell comes out from these protein's work, called as "selective gene expression".

This function expression process of cell is very similar to the JVM bytecode program execution process from the perspective of program. Firstly, the bytecode sequence stored in the peripheral storage or memory is loaded into the JVM, resolves to the class's bytecode, making it into the running status, then the running status of bytecode sequences is translated into the machine code sequence to perform specific tasks. According to the needs of the running environment, the program dynamically loaded follow-up bytecode sequences. The DNA coding program stored in the living cells completed two tasks at the same time, the first of which is to determine when and how the cell would divide, while the second is to determine how to express the cell's function after the cell division. From the perspective of the program, it is easy to understand this work, similar to the von Neumann computer system's basic principle: "Stored Procedures, sequential execution." The chromosome in the cell stores the life program coding sequence. In the process of cell division and differentiation, it continually loads and decodes the corresponding gene code segments to complete specific functional expression of the cell and accomplish the subsequent differentiation regulation.

Modeling and Simulation of Gene Class and Chromosome Class

Because the basic unit of the genome consisting of genes, a Java class is applied to simulate a gene. After summary and abstract of genes, a gene consists of three sub-coded components: the part of cell specific function expression, the part of regulating cell division and differentiation, as well as other auxiliary coding part. The former two parts are the key coding, abstracting gene structure and some source code shown in Figure 4. The genome is composed of a group of such basic genes combination, shown in Figure 5. To make it easier understand, Figure 6 shows the program code sequence from different perspectives.

The paper is mainly focused on the gene coding sequence, and there is no concern of chromosome's spatial structure. It can be simply thought that an organism's chromosome is a linear storage from the point of coding sequence's view. (In fact, from the point of view of the computer, chromosome should be a kind of compressed space storage structure, corresponding to the compression coding technology of the computer.)

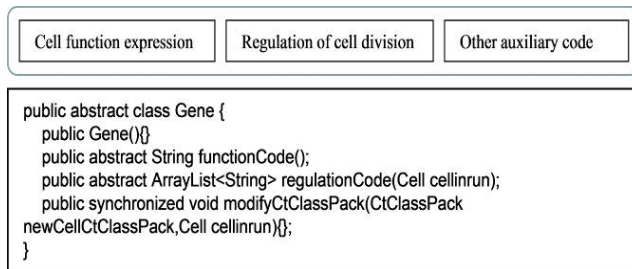


FIGURE 4 GENE ABSTRACT

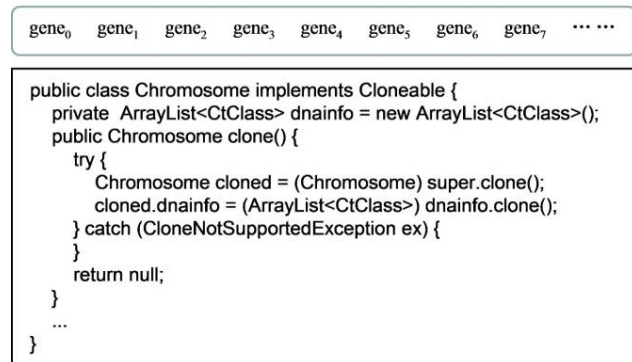


FIGURE 5 CHROMOSOME ABSTRACT

Gene ₀	Gene ₁	Gene ₂		Gene _n	Genomic perspective
...ATTG...	...TTGC...	...CCTT...	...AATTG...	...GGCC...	ATGC coding perspective
...010010...	...010100...	...101110...	...11110110...	...100111...	Binary coding perspective
Gene0.class	Gene1.class	Gene2.class		Genen.class	Java class files perspective

FIGURE 6 ABSTRACT FROM A DIFFERENT PERSPECTIVE ON GENOME

The Abstract of Cell Class and the Program Interpretation of the Cell Functional Expression

In this paper, the program model abstraction is mainly used to simulate the cell function expression process rather than to demonstrate the real life, so the cell class has been abstracted simply in order to seize the essence of the problem. It has been known that every cell is enclosed by cytomembrane and assembly with the cytoplasm and other organelles, and this is consistent with the object-oriented principle of encapsulation, so the cell class simplifies abstract code shown below:

```

public class Cell implements Cloneable {
    Cytoplasm cytoplasm;
    Membrane membrane;
    Nucleus nucleus;
    ...
}

```

The program models made some simplified processing technology, such as the chromosome object suitable for computer storage processing, using a linear list to store the genes class object, so that we can simply locate a particular gene through an index. These simplifications just improve the program's efficiency and reduce programming complexity. By

combining the Java Reflection technology (Brian,1982) and the Bytecode engineering technology (Shigeru, 2004) the paper designs the program model to simulate the cell function expression process. Based on the abstract above, the simulation process is put forward as follows:

Firstly, appropriate gene class was selected from the stored code sequence and in this case javassist.CtClass (an abstract representation of a Java class in non-running state) object has been utilized. This is an abstract, non-operational status of the object of a Java class coding sequence that corresponds to the unwinding and selected from the chromosomal gene fragment.

Secondly, the gene class objects were loaded into JVM through custom classloader, which made it into running state. Here java.lang.Class (an abstract representation of a Java class in running state) object was used. This procedure simulated the transcription from DNA to RNA. In this procedure the coding sequence was not changed, but transformed from stored status into running status.

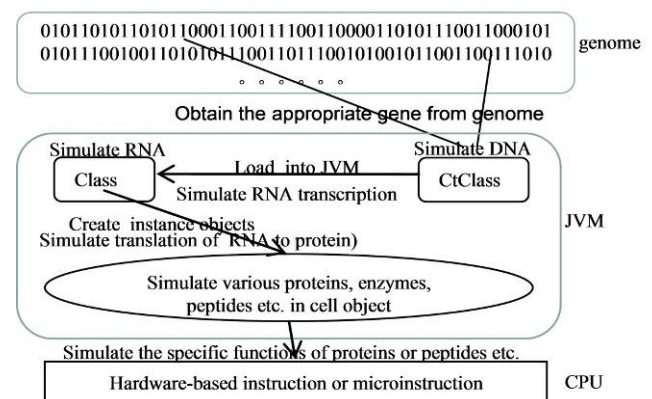


FIGURE 7 THE PRINCIPLE DIAGRAM OF THE FUNCTION OF GENE EXPRESSION IN CELLS

Thirdly, through dynamic reflection technology, Java. Lang. Class object was used to create specific cell instance objects, and then the corresponding function called code (corresponding to the local machine instruction) was generated, which simulated the translation from RNA to protein. Finally, the instance object executed the corresponding instruction sequence, and simulated the specific function expression of a cell, corresponding to the protein that controls various metabolic chemical reactions and transport. The Schematic diagram of the abstract program model is shown in Figure 7.

Conclusions

In this paper, the similarities of DNA coding sequence

and Java bytecode program coding sequence have been compared and analyzed, then the common characteristics were extracted from the principle of the program executing that is the coding sequence from stored status turned into the macroscopic phenomena which can be perceived. Then the program working principle was employed to understand the working mechanism of cells and it was found that the structure composition and function of the cell are very suitable for object-oriented manner modeling. Some program models were abstracted and designed to simulate the cellular function expression, in which the focus was on the abstraction of the class of gene, chromosome, cell, cytoplasm, cell nucleus and cell membrane etc., and the gene class and the cell class are the key to all program models. The program models can well simulate and demonstrate how an explicit function expression of the cell was transformed from DNA coding sequence into RNA coding sequence, then into proteins, and finally the proteins involve in various of chemical reaction, transport or form tissue and show the specific function of the cell.

In the process of the program models simulation, this paper assumes that a cell is a control unit, similar to the CPU in the computer. The cell membrane represents the boundary of the input and output, the cytoplasm similar memory, bus and other internal circuits, the mitochondria in the cytoplasm is power primary supply equipment, and the chromosomes in the nucleus is the program storage devices of life program, etc. From the program models, it can be seen that the life program working mechanism in cells is similar to the Java program working mechanism in computers, the DNA coding sequence corresponding to an ordered set of JVM instructions stored status, the RNA coding sequence corresponding to the running status, and protein coding sequence corresponding to the machine instructions and data which the local CPU can identify.

REFERENCES

- Bill Gates, "The Road Ahead", Penguin Books, London, p228, 1997.
- Brian C. Smith. "Reflection and Semantics in a Procedural Language", Ph.D. Thesis. Massachusetts Institute of Technology. MIT-LCS-TR-272. January 1982.
- Brown T.A., "Genomes", John Wiley & Sons, 1999.
- Crick F.H.C., "On Protein Synthesis", Symp. Soc. Exptl. Biol. 1958.
- Cristianini NaH, M., "Introduction to Computational Genomics", Cambridge University Press, 2006.
- Jun M., Shuyan L., Yide M., "Programming Hypothesis on Life Phenomena and the Key Processes Simulation", Advanced Materials Research Vol. 647 (2013) pp258-263, Trans Tech Publications, Switzerland. doi: 10.4028/www.scientific.net/AMR.647.258, 2013.
- Mount D. "Bioinformatics, Sequence and Genome Analysis", pp2-3, Cold Spring Harbor Laboratory Press, 2000.
- Richard Dawkins, "River out of Eden: A Darwinian View of Life", New York: Basic, p17, 1995.
- Shigeru Chiba, "Javassist: Java Bytecode Engineering Made Simple", Java Developer's Journal, vol.9, issue 1, January 8, 2004.
- Steiner R.F., "The Chemical Foundations of Molecular Biology", Princeton: Van Nostrand Comp, 1965.
- Stenesh J., "Dictionary of Biochemistry and Molecular Biology", New York: John Wiley & Sons. 1989.
- Watson, J.D., "Molecular Biology of the Gene[M]", Third Edition, W.A. Benjamin, Inc., 1976.



Jun Ma, Lanzhou City, Gansu Province, China, 1972.10.1

PhD, The Engineer of The School of Information Science and Engineering, Lanzhou University



Mei Fan, Lanzhou City, Gansu Province, China, 1974.7.24

M.E, The Engineer of The School of Information Science and Engineering, Lanzhou University